SIAM J. COMPUT.
Vol. 32, No. 3, pp. 654–670

AN APPROXIMATION ALGORITHM FOR MINIMUM CONVEX COVER WITH LOGARITHMIC PERFORMANCE GUARANTEE*

STEPHAN J. EIDENBENZ† AND PETER WIDMAYER‡

Abstract. The problem MINIMUM CONVEX COVER of covering a given polygon with a minimum number of (possibly overlapping) convex polygons is known to be *NP*-hard, even for polygons without holes [J. C. Culberson and R. A. Reckhow, *J. Algorithms,* 17 (1994), pp. 2–44]. We propose a polynomial-time approximation algorithm for this problem for polygons with or without holes that achieves an approximation ratio of $O(\log n)$, where n is the number of vertices in the input polygon. To obtain this result, we first show that an optimum solution of a restricted version of this problem, where the vertices of the convex polygons may lie only on a certain grid, contains at most three times as many convex polygons as the optimum solution of the unrestricted problem. As a second step, we use dynamic programming to obtain a convex polygon which is maximum with respect to the number of "basic triangles" that are not yet covered by another convex polygon. We obtain a solution that is at most a logarithmic factor off the optimum by iteratively applying our dynamic programming algorithm. Furthermore, we show that MINIMUM CONVEX COVER is *APX*-hard; i.e., there exists a constant $\delta > 0$ such that no polynomial-time algorithm can achieve an approximation ratio of $1+\delta$. We obtain this result by analyzing and slightly modifying an already existing reduction [J. C. Culberson and R. A. Reckhow, *J. Algorithms,* 17 (1994), pp. 2–44].

Key words. convex cover, art gallery, inapproximability, approximation algorithms, dynamic programming

AMS subject classifications. 68U05, 68W25, 68Q25

PII. S0097539702405139

1. Introduction and problem definition. The problem MINIMUM CONVEX COVER is the problem of covering a given polygon T with a minimum number of (possibly overlapping) convex polygons that lie in T. This problem belongs to the family of classic art gallery problems; it is known to be *NP*-hard for input polygons with holes [17] and without holes [4]. The study of approximations for hard art gallery problems has rarely led to good algorithms or good lower bounds; we discuss a few exceptions below. In this paper, we propose the first nontrivial approximation algorithm for MINIMUM CONVEX COVER. Our algorithm works for polygons with and without holes. It relies on a strong relationship between the continuous original problem version and a particular discrete version in which all relevant points are restricted to lie on a kind of grid that we call a quasi grid. The quasi grid is the set of intersection points of

*Received by the editors April 8, 2002; accepted for publication (in revised form) December 27, 2002; published electronically April 17, 2003. Preliminary versions of this paper have appeared in conference proceedings [in Algorithms—ESA 2001 (Århus), Lecture Notes in Comput. Sci. 2161, Springer-Verlag, Berlin, 2001, pp. 333–343] and in Stephan Eidenbenz's thesis [Ph.D. thesis, Dissertation ETH 13683, Zürich, Switzerland, 2000]. This work was partially supported by the Swiss Federal Office for Education and Science under contract BBW 01.0068 within a project funded by the European Union under contract IST-2001-32007 (APPOL II). This work was performed by an employee of the U.S. Government or under U.S. Government contract. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

http://www.siam.org/journals/sicomp/32-3/40513.html

†P.O. Box 1663, MS M997, Los Alamos National Laboratory, Los Alamos, NM 87545 (eidenben @lanl.gov) LA-UR:02-7641. This author's work was done while he was at the Institute of Theoretical Computer Science, Swiss Federal Institute of Technology, Zürich, Switzerland.

‡Institute of Theoretical Computer Science, Swiss Federal Institute of Technology, 8092 Zürich, Switzerland (widmayer@inf.ethz.ch).

all lines connecting two vertices of the input polygon. In the Restricted Minimum Convex Cover problem, the vertices of the convex polygons that cover the input polygon may lie only on this quasi grid. We prove that an optimum solution of the Restricted Minimum Convex Cover problem needs at most three times the number of convex polygons that the Minimum Convex Cover solution needs. To find an approximate solution for the Restricted Minimum Convex Cover problem, we propose a greedy approach: we compute one convex polygon of the solution after the other, and we pick as the next convex polygon one that covers a maximum number of triangles defined on an even finer quasi grid, where these triangles are not yet covered by previously chosen convex polygons. We propose an algorithm for finding such a maximum convex polygon by means of dynamic programming. To obtain an upper bound on the quality of the solution, we interpret our covering problem on triangles as a special case of the general Minimum Set Cover problem that gives as input a base set of elements and a collection of subsets of the base set and that asks for a smallest number of subsets in the collection whose union contains all elements of the base set. In our special case, each triangle is an element, and each possible convex polygon is a possible subset in the collection, but not all of these subsets are represented explicitly. (There could be an exponential number of subsets.) This construction translates the logarithmic quality of the approximation from Minimum Set Cover to Minimum Convex Cover [13].

On the negative side, we show that Minimum Convex Cover is *APX*-hard; i.e., there exists a constant $\delta > 0$ such that no polynomial-time algorithm can achieve an approximation ratio of $1 + \delta$ (see [3] for an introduction to the class *APX*). This inapproximability result is based on a problem transformation shown by Culberson and Reckhow [4]; we modify this transformation and show that it is gap-preserving (as defined by Arora and Lund [1]).

The related problem of partitioning a given polygon into a minimum number of nonoverlapping convex polygons is polynomially solvable for input polygons without holes [2]. It is *NP*-hard for input polygons with holes [15] and can be approximated with an approximation ratio of 4 [12]; it remains *NP*-hard even if the convex partition must be created by cuts from a given family of (at least three) directions [16]. Other related results for art gallery problems include approximation algorithms with logarithmic approximation ratios for Minimum Vertex Guard and Minimum Edge Guard [10], as well as for the problem of covering a polygon with rectangles in any orientation [11]. Furthermore, logarithmic inapproximability results are known for Minimum Point/Vertex/Edge Guard for polygons with holes, and *APX*-hardness results are known for the same problems for polygons without holes [6]. The related problem Rectangle Cover of covering a given orthogonal polygon with a minimum number of rectangles can be approximated with a constant ratio for polygons without holes [9] and with an approximation ratio of $O(\sqrt{\log n})$ for polygons with holes [14]. For additional results, see the surveys on art galleries [18, 19]. The general idea of using dynamic programming to find maximum convex structures has been used before to solve the problem of finding a maximum (with respect to the number of vertices) empty convex polygon, given a set of vertices in the plane [5]. An $O(\log n)$ approximation algorithm for the problem of covering a polygon with rectangles in any orientation [11] relies on an approach similar to ours that consists of making the problem discrete and then transforming it to a Minimum Set Cover instance.

In section 2, we define the quasi grid and its refinement into triangles. Section 3

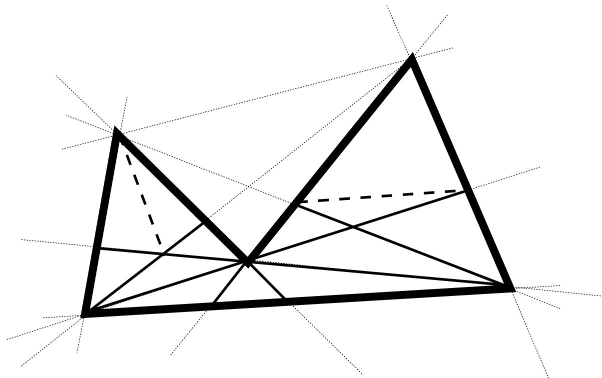

FIG. 1. *Construction of first-order basic triangles.*

contains the proof of the linear relationship between the sizes of the optimum solutions of the unrestricted and restricted convex cover problems. We propose a dynamic programming algorithm to find a maximum convex polygon in section 4 before showing how to iteratively apply this algorithm to find a convex cover in section 5. In section 6, we present the proof of the APX-hardness of MINIMUM CONVEX COVER. Concluding thoughts are in section 7.

2. From the continuous to the discrete. We consider simple input polygons with and without holes, where a polygon T is given as an ordered list of vertices in the plane. If T contains holes, each hole is also given as an ordered list of vertices. Let V_T denote the set of vertices (including the vertices of holes, if any) of a given polygon T. While in the general MINIMUM CONVEX COVER problem the vertices of the convex polygons that cover the input polygon can be positioned anywhere in the interior or on the boundary of the input polygon, we restrict their positions in an intermediate step: they may be positioned only on a quasi grid in the RESTRICTED MINIMUM CONVEX COVER problem.

In order to define the RESTRICTED MINIMUM CONVEX COVER problem more precisely, we partition the interior of a polygon T into *convex components* (as proposed in [10] for a different purpose) by drawing a line through each pair of vertices of T. We then triangulate each convex component arbitrarily. We call the triangles thus obtained *first-order basic triangles.* Figure 1 shows an example of the first-order basic triangles of a polygon (thick solid lines) with an arbitrary triangulation (fine solid lines and dashed lines). If a polygon T consists of n vertices, drawing a line through each pair of vertices of T will yield less than $\binom{n}{2} \cdot \binom{n}{2} \in O(n^4)$ intersection points. Let V_T^1 be the set of these intersection points that lie in T (in the interior or on the boundary). Note that $V_T \subseteq V_T^1$. The first-order basic triangles are a triangulation of V_T^1 inside T; therefore, the number of first-order basic triangles is also $O(n^4)$. The RESTRICTED MINIMUM CONVEX COVER problem asks for a minimum number of convex polygons, with vertices restricted to V_T^1, that together cover the input polygon T. We call V_T^1 a quasi grid that is imposed on T. For solving the RESTRICTED MINIMUM CONVEX COVER problem, we make use of a finer quasi grid: simply partition T by drawing lines through each pair of points from V_T^1. This yields again convex components, and we triangulate them again arbitrarily. This higher resolution partition yields $O(n^{16})$ intersection points, which define the set V_T^2. We call the resulting triangles *second-order basic triangles.* Obviously, there are $O(n^{16})$ second-order basic triangles. Note

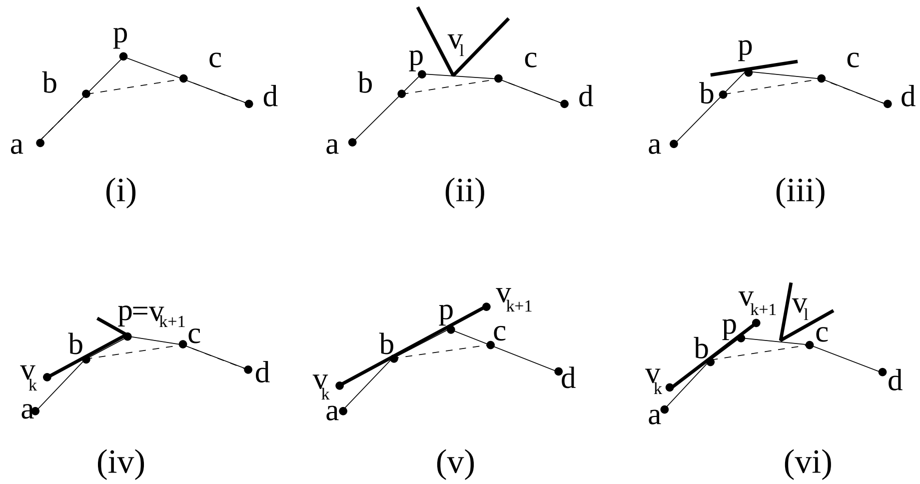

FIG. 2. *Expansion of edge (b,c).*

that $V_T \subseteq V_T^1 \subseteq V_T^2$.

3. The optimum solution of MINIMUM CONVEX COVER vs. the optimum solution of RESTRICTED MINIMUM CONVEX COVER. The quasi grids V_T^1 and V_T^2 serve the purpose of making a convex cover computationally efficient while at the same time guaranteeing that the cover on the discrete quasi grid is not much worse than the desired cover in continuous space. The following theorem proves the latter.

THEOREM 1. *Let T be an arbitrary simple input polygon with n vertices. Let OPT denote the size of an optimum solution of* MINIMUM CONVEX COVER *with input polygon T, and let OPT' denote the size of an optimum solution of* RESTRICTED MINIMUM CONVEX COVER *with input polygon T. Then*

$$OPT' \leq 3 \cdot OPT.$$

Proof. We proceed as follows: we show how to *expand* a given arbitrary convex polygon $C \subseteq T$ to another convex polygon $C' \subseteq T$ with $C \subseteq C'$ by iteratively expanding edges. We then replace the vertices in C' by vertices from V_T^1, which results in a (possibly) nonconvex polygon $C'' \subseteq T$ with $C' \subseteq C''$. Finally, we describe how to obtain three convex polygons C_1'', C_2'', C_3'' with $C'' = C_1'' \cup C_2'' \cup C_3''$ that contain only vertices from V_T^1. This will complete the proof, since each convex polygon from an optimum solution of MINIMUM CONVEX COVER can be replaced by at most three convex polygons that are in a solution of RESTRICTED MINIMUM CONVEX COVER. Following this outline, let us present the proof details.

Expanding edges. Let C be an arbitrary convex polygon inside polygon T. Let the vertices of C be given in clockwise order. We obtain a series of convex polygons $C^1, C^2, \ldots, C'$ with $C = C^0 \subseteq C^1 \subseteq C^2 \subseteq \cdots \subseteq C'$, where C^{i+1} is obtained from C^i as follows (see Figure 2).

Let a, b, c, d be consecutive vertices (in clockwise order) in the convex polygon C^i that lies inside polygon T. For ease of description, we assume that C^i does not contain vertices that are collinear with its two neighboring vertices, except when such a vertex happens to be a vertex from V_T; moreover, any vertex from V_T that lies on the boundary of C^i is also a vertex of C^i, even if it has collinear neighbors in C^i.

Let vertices $b, c \notin V_T$, with b and c not on the same edge of T. Then the edge (b, c) is called *expandable*. If there exists no expandable edge in C^i, then $C' = C^i$, which means that we have found the end of the series of convex polygons. If (b, c) is an expandable edge, we *expand* the edge from vertex b to vertex c as follows:

- If b does not lie on the boundary of T, then we let a point p start at b and move along the halfline through a and b away from a and b until either one of the following two events happens: p lies on the line through c and d, or the triangle p, c, b touches the boundary of T. Fix p as soon as the first of these events happens. Figure 2 shows a list of all possible cases, where the edges from polygon T are drawn as thick edges: point p lies on the intersection point of the lines from a through b and from c through d as in case (i), or there is a vertex v_l on the line segment from p to c as in case (ii), or p lies on an edge of T as in case (iii).
- If b lies on the boundary of T, i.e., on some edge of T, say, from v_k to v_{k+1} (in clockwise order), then let p move from b as before, except that the move is now along the halfline from v_k through b away from v_k and b up until at most v_{k+1} (instead of the ray from a through b). Figure 2 shows a list of all possible cases: point p lies either at vertex v_{k+1} as in case (iv) or on the intersection point of the lines from b to v_{k+1} and from d through c as in case (v), or there is a vertex v_l on the line segment from p to c as in case (vi).

A new convex polygon C_p^i is obtained by simply adding point p as a vertex in the ordered set of vertices of C^i between the two vertices b and c; if—as in cases (ii) and (vi)—a vertex from V_T lies on the boundary of C_p^i, it is also added as a vertex (despite the fact that it may have two collinear neighbors). In contrast, all vertices in C_p^i that have collinear neighbors and that are not vertices in V_T are eliminated.

An edge from two consecutive vertices b and c with $b, c \notin V_T$ can always be expanded in such a way that the triangle b, p, c that is added to the convex polygon is nondegenerate, i.e., has nonzero area, unless b and c both lie on the same edge of polygon T. This follows from the cases (i)–(vi) of Figure 2.

Let $C^{i+1} = C_p^i$ if either a new vertex of V_T has been added to C_p^i in the expansion of the edge, which is true in cases (ii), (iv), and (vi), or the number of vertices of C_p^i that are not vertices from V_T has decreased, which is true in case (i). If p is as in case (iii), we expand the edge (p, c), which will result in case (iv), (v), or (vi). Note that in cases (iv) and (vi), we have found C^{i+1}. If p is as in case (v), we expand the edge (p, d), which will result in case (iv), (v), or (vi). If it is case (v) again, we repeat the procedure by expanding the edge from p and the successor (clockwise) of d. This needs to be done at most as many times as there are vertices in C^i, since the procedure eliminates a vertex from C^i in each iteration and will stop before it tries to expand an edge ending at vertex a as the resulting polygon would not be convex. Therefore, we obtain C^{i+1} from C^i in a finite number of steps.

Let τ_i denote the number of vertices in C^i that are also vertices in T, and let $\hat{\tau}_i$ be the number of vertices in C^i that are not vertices in T. Note that $\phi(i) = \hat{\tau}_i - 2\tau_i + 2n$ is a function that bounds the number of remaining iteration steps that are needed to reach C'; it strictly decreases with every increase in i and cannot become negative as $\hat{\tau}_i$ and τ_i are both nonnegative numbers by definition and $n \geq \tau_i$. The existence of such a bounding function, which is often called a variant function, implies the finiteness of the series $C^1, C^2, \ldots, C'$ of convex polygons.

By definition, there are no expandable edges left in C'. Call a vertex of C' a T-vertex if it is a vertex in T. From the definition of expandable edges, it is clear that

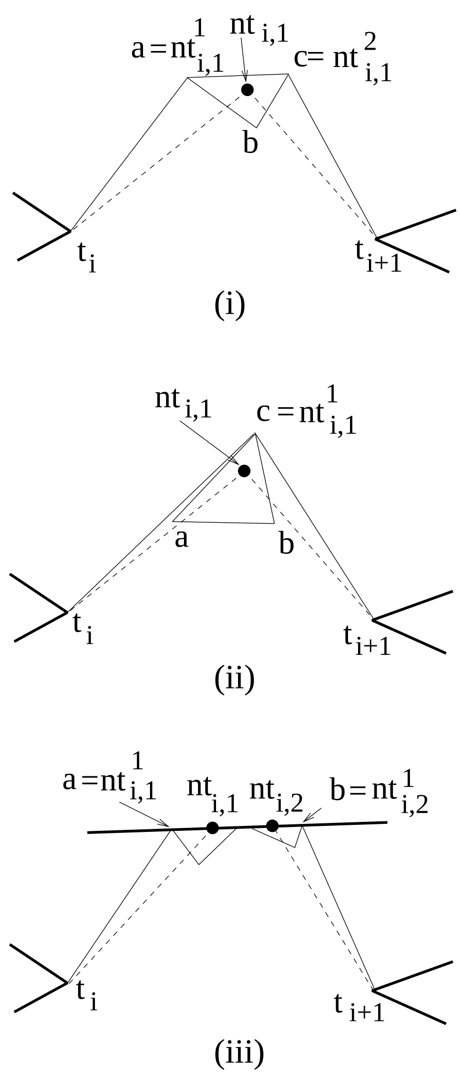

Fig. 3. *Replacing non-T-vertices.*

there can be at most two non-T-vertices between any two consecutive T-vertices in C', and if there are two non-T-vertices between two consecutive T-vertices, they must both lie on the same edge in T. (Otherwise, the edge between two non-T-vertices would be expandable, which contradicts the definition of C'.)

Replacing vertices. Let the T-vertices in C' be $t_1, \ldots, t_l$ in clockwise order, and let the non-T-vertices between t_i and t_{i+1} be $nt_{i,1}$ and $nt_{i,2}$ if they exist. We will replace each non-T-vertex $nt_{i,j}$ in C' by one or two vertices $nt^1_{i,j}$ and $nt^2_{i,j}$ that are both elements of the quasi grid V^1_T. This will transform the convex polygon C' into a not necessarily convex polygon C''. (We will show later how C'' can be covered by at most three convex polygons C''_1, C''_2, C''_3.) The details are as follows: let a, b, c be the first-order basic triangle in which non-T-vertex $nt_{i,j}$ lies, as illustrated in Figure 3. Points a, b, c are all visible from both vertices t_i and t_{i+1}. To see this, assume by contradiction that the view from, say, t_i to a is blocked by an edge e of T. Since $nt_{i,j}$

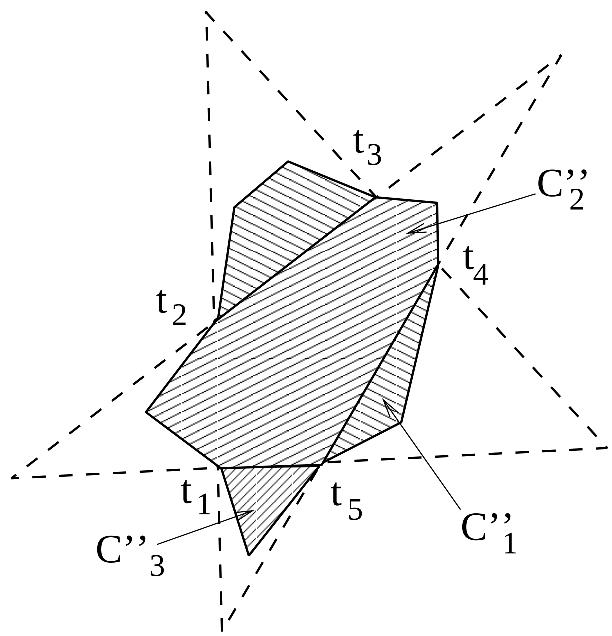

FIG. 4. *Covering C'' with three convex polygons.*

must see t_i, the edge e must contain a vertex e' in the triangle $t_i, a, nt_{i,j}$, but then a cannot be a vertex of the first-order basic triangle in which $nt_{i,j}$ lies, since the line from vertex t_i through vertex e' would cut through the first-order basic triangle—an impossibility.

Assume that only one non-T-vertex $nt_{i,1}$ exists between t_i and t_{i+1}. If the triangle t_i, t_{i+1}, a completely contains the triangle $t_i, nt_{i,1}, t_{i+1}$, then we let $nt^1_{i,1} = a$, and likewise for b and c (see Figure 3 (ii)). Otherwise, we let $(nt^1_{i,1}, nt^2_{i,1})$ be $(a, b), (a, c)$, or (b, c), as in Figure 3 (i), such that the polygon $t_i, nt^1_{i,1}, nt^2_{i,1}, t_{i+1}$ is convex and completely contains the triangle $t_i, nt_{i,1}, t_{i+1}$. This is always possible by the definition of points a, b, c.

Assume that two non-T-vertices $nt_{i,1}$ and $nt_{i,2}$ exist between t_i and t_{i+1}. From the definition of C', we know that $nt_{i,1}$ and $nt_{i,2}$ must lie on the same edge e of T. Therefore, the basic triangle in which $nt_{i,1}$ lies must contain a vertex a either at $nt_{i,1}$ or preceding $nt_{i,1}$ on edge e along T in clockwise order. Let $nt^1_{i,1} = a$. The basic triangle in which $nt_{i,2}$ lies must contain a vertex b either at $nt_{i,2}$ or succeeding $nt_{i,2}$ on edge e. Let $nt^1_{i,2} = b$. (See Figure 3 (iii).) Note that the convex polygon $t_i, nt^1_{i,1}, nt^1_{i,2}, t_{i+1}$ completely contains the polygon $t_i, nt_{i,1}, nt_{i,2}, t_{i+1}$.

After applying this change to all non-T-vertices in C', we obtain a (possibly) nonconvex polygon C''.

Covering with three convex polygons. We will now show how to cover C'' with at most three convex polygons. First, assume that C'' contains an odd number f of T-vertices. We let C''_1 be the polygon defined by vertices $t_i, nt^k_{i,j}$, and t_{i+1} for all j, k and for all odd i, but $i \neq f$. By construction, C''_1 is convex. To see this, assume C''_1 is not convex; it would then have to have at least one vertex whose inner angle is larger than π, which cannot happen at non-T-vertices in C''_1 by construction. The inner angles at a T-vertex t_{i-1} for i odd cannot be larger than π either, because

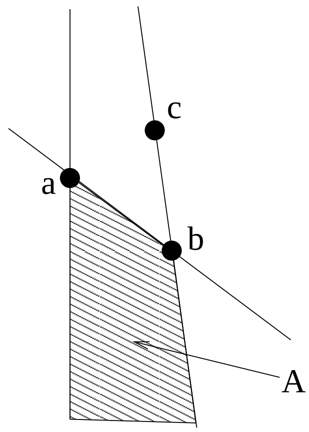

Fig. 5. *Dynamic programming.*

polygon C_1'' lies entirely to the right of the line going from t_{i-1} through t_i. Let C_2'' be the polygon defined by vertices $t_i, nt_{i,j}^k$, and t_{i+1} for all j, k and for all even i. Finally, let C_3'' be the polygon defined by vertices $t_f, nt_{f,j}^k$, and t_1 for all j, k. Using similar arguments as for C_1'', polygons C_2'' and C_3'' are convex as well. Figure 4 shows an example. Obviously, C_1'', C_2'', and C_3'' together cover all of C''. Second, assume that C'' contains an even number of T-vertices, and cover it with only two convex polygons using the same concept. This completes the proof. ∎

4. Finding maximum convex polygons. Assume that each second-order basic triangle from a polygon T is assigned a weight value of either 1 or 0. In this section, we present an algorithm using dynamic programming that computes a convex polygon M in a polygon T that contains a maximum number of second-order basic triangles with weight 1 and that has vertices only from V_T^1. For simplicity, we call such a polygon a *maximum convex polygon*. The weight of a polygon M is defined as the sum of the weights of the second-order basic triangles in the polygon and is denoted by $|M|$. We will later use the algorithm described below to iteratively compute a maximum convex polygon with respect to the triangles that are not yet covered, to eventually obtain a convex cover for T.

Let $a, b, c \in V_T^1$. Let $P_{a,b,c}$ denote the maximum convex polygon that
- contains only vertices from V_T^1,
- contains vertices a, b, c in counterclockwise order,
- has a as its left-most vertex,[1]
- contains additional vertices only between vertices a and b, and
- is completely contained in T.

Given three vertices $a, b, c \in V_T^1$, let A be the (possibly infinite) area of points that are
- to the right of vertex a,
- to the left of the line oriented from b through a, and
- to the left of the line oriented from b through c.

[1]If polygon $P_{a,b,c}$ has several left-most vertices, vertex a is one of them.

Initialize table $S(a,b,c)$ with zeros	
FORALL $a \in V_T^1$ DO	$O(n^{28})$
Choose a helper point a' with the same x-coordinate and an arbitrary but smaller y-coordinate than a	
Order all vertices $b \in V_T^1$ to the right of a according to the angle formed by b, a, a'; let the resulting ordered set be B	$O(n^4 \log n)$
$B' := \emptyset$	
WHILE $B \neq \emptyset$ DO	$O(n^{24})$
Let b be the smallest element in B; $B := B - \{b\}$; $B' := B' \cup \{b\}$	
FORALL $c \in V_T^1 \backslash B'$ to the right of a DO	$O(n^{20})$
Compute $\lvert\Delta a,b,c\rvert$	$O(n^{16})$
Define area A with respect to vertices a, b, c according to Lemma 2	
FORALL $d \in (V_T^1 \cap A)$ DO	$O(n^4)$
Look up $\lvert P_{a,d,b}\rvert$ and store maximizing d in $d_{\max}$	
END	
$\lvert P_{a,b,c}\rvert := \lvert\Delta a,b,c\rvert + \lvert P_{a,d_{\max},b}\rvert$	
Store $\lvert P_{a,b,c}\rvert$ in table S	
END	
END	
END	
Find maximum entry in table S	

FIG. 6. *Algorithm for computing a maximum weight convex polygon.*

For an illustration, see Figure 5. Let

$$P'_{a,b,c} = \max_{d \in V_T^1 \cap A} P_{a,d,b} \cup \Delta a,b,c,$$

where $\Delta a,b,c$ is the triangle a,b,c and max is defined as follows (to simplify notation):

$$\max\{P_1, P_2\} = \begin{cases} P_1 & \text{if } |P_1| \geq |P_2|, \\ P_2 & \text{otherwise.} \end{cases}$$

LEMMA 2. *$P_{a,b,c} = P'_{a,b,c}$ if the triangle a,b,c is completely contained in the polygon T.*

Proof. Consider $P_{a,b,c}$, which is maximum by definition. $P_{a,b,c}$ must contain additional vertices between a and b. (Otherwise, the lemma is trivially true.) Let d' be the predecessor of b in the counterclockwise order of $P_{a,b,c}$. Vertex d' must lie in A as defined above. Now consider $P'' = P_{a,b,c} - \Delta a,b,c$. From the definition of A it is clear that P'' can contain only vertices that lie in A. Now $P_{a,d',b}$ is maximum by definition, and it is considered when computing $P'_{a,b,c}$. □

Let M be a maximum convex polygon for a polygon T with weights assigned to the second-order basic triangles. Let a be the left-most vertex of M, let c be the predecessor of a in M in counterclockwise order, and let b be the predecessor of c. Then $|P_{a,b,c}| = |M|$ by definition. We will use Lemma 2 to construct an algorithm, which takes as input a polygon T and an assignment of weight 0 or 1 to each second-order basic triangle of T and computes the maximum convex polygon. An overview of the algorithm is given in Figure 6.

In more detail, we start by initializing a table $S(a, b, c)$, where the entry at position a, b, c denotes the weight $|P_{a,b,c}|$, in line 1 of Figure 6. In a first loop, we fix vertex $a \in V_T^1$ in line 2, let a' be a helper point with the same x-coordinate and an arbitrary but smaller y-coordinate than a, and order all vertices $b \in V_T^1$ to the right of a according to the angle formed by b, a, a'. We call the resulting ordered set B and let B' be the empty set. In a second loop, starting at line 6, we iteratively take the smallest element b from B, remove it from B, and add it to set B'; then for every $c \in V_T^1 \backslash B'$ to the right of a (see line 8), we compute weight $|\Delta a, b, c|$ of the triangle a, b, c and compute $P_{a,b,c}$ according to Lemma 2 in line 11 (i.e., look up the values of $P_{a,d,b}$ for all $d \in V_T^1 \cap A$ and take the maximum; all these values were computed in earlier iterations). We then compute weight $|P_{a,b,c}|$ by adding $|\Delta a, b, c|$ to $|P_{a,d,b}|$, where d is the maximizing argument, and store the value in table S. Note that the computation of $P_{a,b,c}$ according to Lemma 2 is always possible, since all possible vertices d in $P_{a,d,b}$ lie to the left of the line from b to a (see also definition of area A), have therefore smaller angles d, a, a' than b, a, a', and have therefore already been computed. The algorithm is executed for every $a \in V_T^1$, and—by using standard bookkeeping techniques (not explicitly given in the pseudocode of Figure 6)—the maximum convex polygon found is returned.

The cumulative running times of the loops and the running times of some crucial individual lines of the algorithm are given in Figure 6, resulting in an overall running time of $O(n^{28})$. To see this, we first look at the loop from line 8 to line 16: each iteration of this loop takes time $O(n^{16})$, which is the running time of computing the weight of a triangle a, b, c (see line 9) as we have to add the weights of almost all second-order basic triangles; the $O(n^4)$ running time of the inner loop (lines 11 to 13) is dominated by the $O(n^{16})$ running time of line 9. Since there are $O(n^4)$ iterations of the loop from line 8 to line 16, we get a running time of $O(n^{20})$ for this loop. The loop from lines 6 to 17 consists of a total of $O(n^4)$ iterations of the $O(n^{20})$ loop from lines 8 to 16, thus resulting in a cumulative running time of $O(n^{24})$. Finally, the loop from lines 2 to 18 has $O(n^4)$ iterations of the $O(n^{24})$ loop from lines 6 to 17; the $O(n^4 \log n)$ time required for sorting in line 4 is dominated by the $O(n^{24})$ time for the loop. Thus the overall running time is $O(n^{28})$. Memory requirements are $O(n^{12})$ as we need to allocate table S.

5. An approximation algorithm for Minimum Convex Cover. Given a polygon T, we obtain a convex cover by iteratively applying the algorithm for computing a maximum convex polygon from section 4. It works as follows for an input polygon T:

1. Let all second-order basic triangles have weight 1. Let $S = \emptyset$.
2. Find the maximum convex polygon M of polygon T using the algorithm from section 4, and add M to the solution S. Decrease the weight of all second-order basic triangles that are contained in M to 0.[2]
3. Repeat step 2 until there are no second-order basic triangles with weight 1 left. Return S.

To obtain a performance guarantee for this algorithm, consider the Minimum Set Cover instance I, which has all second-order basic triangles as elements and where the second-order basic triangles with weight 1 of each convex polygon in T, which contains only vertices from V_T^1, form a set in I. The greedy heuristic for Minimum

[2]Note that by the definition of second-order basic triangles, a second-order basic triangle either is completely contained in M or is completely outside M.

Set Cover achieves an approximation ratio of $1 + \ln n'$, where n' is the number of elements in I [13], and it works in exactly the same way as our algorithm. However, we do not have to (and could not afford to) compute all the sets of the Minimum Set Cover instance I (which would be a number exponential in n'); it suffices to always compute a set, which contains a maximum number of elements not yet covered by the solution thus far. This is achieved by reducing the weights of the second-order basic triangles already in the solution to 0; i.e., a convex polygon with maximum weight is such a set.

Note that $n' = O(n^{16})$ since the number of triangles in a triangulation is proportional to the number of points in V_T^2 that induce the triangulation. Therefore, our algorithm achieves an approximation ratio of $O(\log n)$ for Restricted Minimum Convex Cover on input polygon T. Because of Theorem 1, we know that the solution found for Restricted Minimum Convex Cover is also a solution for the unrestricted Minimum Convex Cover that is at most a factor of $O(\log n)$ off the optimum solution.

As for the running time of this algorithm, observe that the algorithm adds to the solution in each round a convex polygon with nonzero weight. An optimum solution would consist of at most $O(n)$ convex polygons, since a triangulation of the vertices of the input polygon yields a trivial solution with $O(n)$ convex polygons that are triangles in this case. Since our algorithm finds a solution that is at most a factor $O(\log n)$ off the optimum solution and since it adds a convex polygon to the solution in each round, there can be at most $O(n \log n)$ rounds before the algorithm finishes. As each round takes time $O(n^{28})$, the total running time is $O(n^{29} \log n)$. This completes the proof of our first main theorem:

Theorem 3. Minimum Convex Cover *for input polygons with or without holes can be approximated by a polynomial-time algorithm with an approximation ratio of* $O(\log n)$*, where* n *is the number of polygon vertices.*

6. *APX*-hardness of Minimum Convex Cover.

The upper bound of $O(\log n)$ on the approximation ratio for Minimum Convex Cover may not be tight: we will now prove that there is a constant lower bound on the approximation ratio, and hence a gap remains. More precisely, we prove Minimum Convex Cover to be *APX*-hard. Our proof of the *APX*-hardness of Minimum Convex Cover for input polygons with or without holes uses a construction similar to the one that is used to prove the *NP*-hardness of this problem for input polygons without holes[4].[3] However, we reduce the problem Maximum 5-Occurrence-3-Sat rather than SATISFIABILITY (SAT) (as done in the original reduction [4]) to Minimum Convex Cover, and we design the reduction to be gap-preserving [1]. Maximum 5-Occurrence-3-Sat is the variant of SAT in which each variable may appear at most five times in clauses and each clause contains at most three literals. Maximum 5-Occurrence-3-Sat is *APX*-complete [1].

The reduction is constructed as follows: for a given instance I of Maximum 5-Occurrence-3-Sat with n variables $x_1, \ldots, x_n$ and m clauses $c_1, \ldots, c_m$, we construct an instance I' of Minimum Convex Cover. To stick to the notation of [4], let $l_i \leq 5$ denote the number of literals of variable x_i in the clauses, and let $l = \sum_{i=1}^{n} l_i$ be the total number of literals.

For each literal in I, we construct a literal pattern, which we call a "beam ma-

[3] *APX*-hardness for Minimum Convex Cover for input polygons without holes implies *APX*-hardness for the same problem for input polygons with holes.

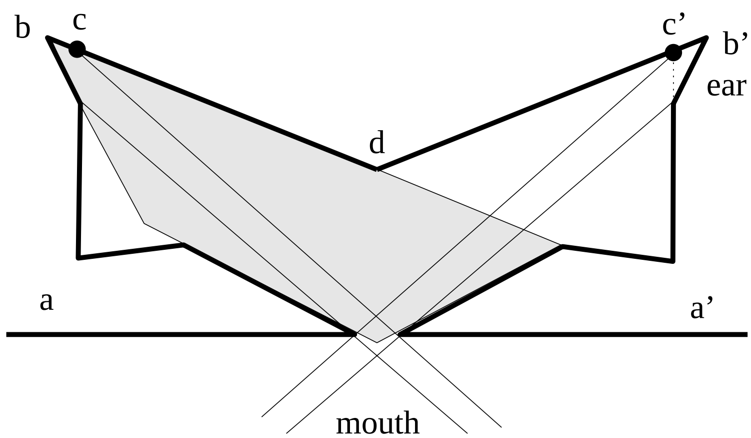

FIG. 7. *The beam machine.*

chine," as illustrated in Figure 7. A beam machine allows us to send a beam, i.e., a slim convex polygon in one of two possible directions out of the beam machine toward a structure that represents a clause. The beam machines of all literals of a variable are then combined into a variable structure, as illustrated in Figure 8. All these variable structures are then arranged in a half-circle such that the beams emitted from the beam machines reach the appropriate clause checkers, which are simple dents. An overview of the whole structure is given in an example in Figure 9. After this overview, let us give a more detailed description.

The beam machine that is constructed for each literal is shown in Figure 7. Since no two of the four vertices a, a', b, and b' see each other, at least four convex polygons are needed to cover the beam machine. Two of these are the maximal convex polygons a, c, d and a', c', d. The remaining areas around the mouth and the ear (the triangle) at b or b' can be covered by a large convex polygon shown in light gray in Figure 7. Finally, a fourth convex polygon is needed to cover the other ear (at b' in Figure 7). This polygon, which we call a beam, is very slim and can be extended indefinitely beyond the mouth outside the beam machine. The large light gray convex polygon thus acts as a switch: depending on whether we let it cover the ear at b or b', we can turn on the indefinite beam polygon at the other ear. However, we cannot turn on both beams and still use only four polygons to cover the beam machine. Note that we can "focus" and "aim" the beam by slightly bending the whole beam machine or by making the ears smaller.

The variable structure is illustrated in Figure 8. Its basic shape is butterfly-like. The beam machines for each occurrence of the variable in a literal in a clause are set on top of the butterfly with the positive literals on the right wing and the negative literals on the left wing of the butterfly. For each literal, we have a dent on the bottom line of the wing. If we cover each dent of the left or right wing with a maximal convex polygon, i.e., with a polygon that covers the whole dent and then extends canonically, then we have covered almost all of the left or right wing except for the area around the mouth of the variable structure and except for a small triangular region for each literal that lies between two dents. These triangles are called beam locks. We can cover the beam locks either by beams emanating from the beam machines or by a single large convex polygon which also covers the region around the mouth of the variable structure. Such a polygon is drawn in light gray in Figure 8. In a similar

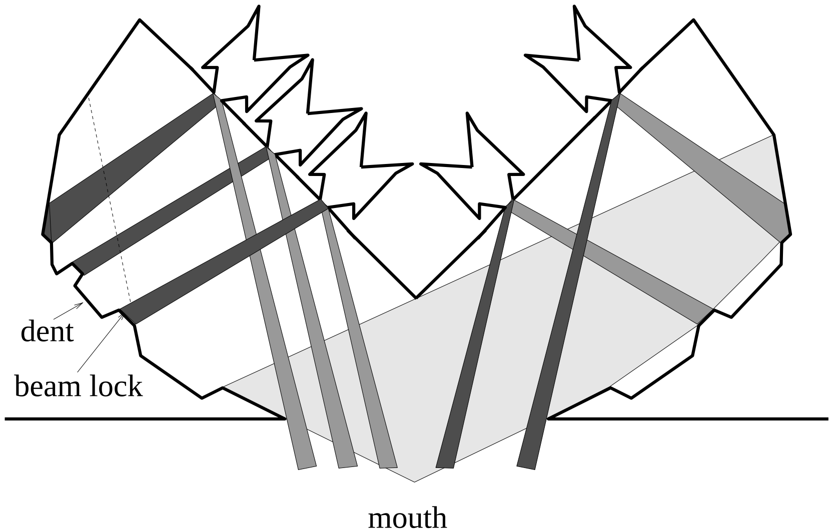

FIG. 8. *The variable structure.*

way as in the beam machine, this large convex polygon acts as a switch: in order to cover the whole variable structure with a minimum number of convex polygons, we can have the beam locks of only one wing covered with such a single polygon; the beam locks of the other wing must be covered by the beams of the beam machines. In Figure 8, beams that are turned on are drawn in dark gray, while beams that are turned off are medium gray. Thus, in Figure 8, all beam machines of positive literals are turned off, and all beam machines of negative literals are turned on and can shine infinitely far beyond the mouth of the variable structure.[4]

We need four convex polygons to cover each beam machine; thus we need $4l_i$ convex polygons to cover the beam machines in the variable structure for variable x_i. For each literal, we need an additional polygon to cover the dent, and we need one additional large switcher polygon to cover the mouth and the beam locks of either the positive or negative literals. Thus a minimum number of $5l_i + 1$ convex polygons are required to cover the variable structure of variable x_i. Note that if the beams of only one negative and one positive literal that are both aimed toward and beyond the mouth of the variable structure are turned on, then $5l_i+2$ convex polygons are needed to cover the variable structure. On the other hand, if the beams of all (positive and negative) literals that cover the beam locks are turned on, there are still $5l_i+1$ convex polygons needed to cover the variable structure, since we also need to cover the area around the mouth.

We arrange all variable structures in a half-circle-like shape above a base line, which contains triangular dents that represent the clauses, as illustrated in Figure 9. This is done in such a way that a beam emanating from a beam machine of a literal that appears in a clause reaches the corresponding dent (the clause checker) that represents that clause and thus covers it. Note that we can arrange the variable structures in such a way that they cannot interfere with each other; i.e., no convex polygon can cover any beam locks or areas around the mouth of two different variable structures. We can achieve this by making the angles at the mouth of each variable

[4]The beam machines have not been drawn exactly to scale in Figure 8.

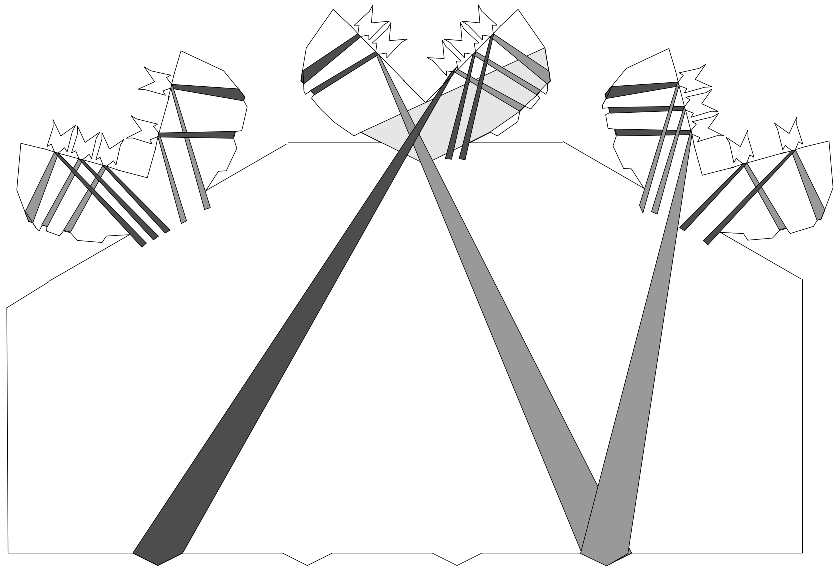

FIG. 9. *Overview of the construction.*

structure very small.

THEOREM 4. *Let I be an instance of* MAXIMUM 5-OCCURRENCE-3-SAT *consisting of n variables and m clauses with a total of l literals, and let I' be the corresponding instance of* MINIMUM CONVEX COVER. *Let OPT be the maximum number of satisfied clauses of I by any assignment of the variables. Let OPT' be the minimum number of convex polygons needed to cover the polygon of I', and let $\epsilon > 0$ be constant. Then*

$$OPT = m \Longrightarrow OPT' = 5l + n + 1,$$
$$OPT < (1 - 15\epsilon)m \Longrightarrow OPT' > 5l + n + 1 + \epsilon n.$$

Proof. The first implication is trivial: if we have a variable assignment that satisfies all variables, we turn on the beams that are aimed toward the clause checkers of all beam machines that represent literals that are satisfied by the assignment. We turn on the beams that are aimed toward the beam locks for all other beam machines. Thus we need $5l_i + 1$ convex polygons to cover the variable structure x_i. If we sum this up over all n variables, we obtain $5l + n$ convex polygons. We need one additional polygon to cover the space between the base line and the variable structures.

Since each clause is satisfied, we must have for each clause checker at least one beam turned on that covers it. Thus the convex polygons as just described cover all of I'.

We prove the second implication by proving its contraposition, i.e., $OPT' \leq 5l + n + 1 + \epsilon n \Longrightarrow OPT \geq (1 - 15\epsilon)m$. To this end, we show how to transform the convex polygons of any solution S' of the MINIMUM CONVEX COVER instance I' in such a way that their total number does not increase and in such a way that a truth assignment of the variables satisfying the desired number of clauses can be “inferred” from the convex polygons.

Suppose we are given a solution S' of the CONVEX COVER instance with $|S'| \leq 5l + n + 1 + \epsilon n$.

By construction, the variable generator for variable x_i must be covered by at least $5l_i+1$ convex polygons. Moreover, by construction, there is no convex polygon, which simultaneously covers a part of a beam lock in any variable generator and a part of a clause checker. There is not even a convex polygon which covers a part of a beam lock and touches the horizontal line, on which the clause checkers lie. Similarly, note that there is no convex polygon which can simultaneously cover a part of an ear of a beam machine and a part of any clause checker, except for the clause checker associated with the beam machine.

Proceed in the following order:

1. Determine which convex polygon in S' covers the midpoint on the line segment between the clause checkers of clause c_1 and c_2. Transform this polygon in such a way that it covers all of the area between the clause checkers and the variable generators. Note that no convex polygon that covers this midpoint can also cover any beam lock, ear of a beam machine, or clause checker. Therefore, we have a feasible solution after this step.
2. For each clause checker, proceed as follows: for each convex polygon in S' that covers part of the clause checker and that is not a regular beam which leads to a beam machine associated with the clause checker, turn the polygon into a beam to any of the associated beam machines.
3. If there exists a convex polygon in S' that covers parts of the interior of at least two different variable structures, then choose any variable structure in which it lies, and cut off all other parts. This operation results in a feasible solution since, by construction, such a polygon cannot cover the beam locks or the area around the mouths of two different variable structures.
4. For each variable structure, proceed as follows:
   - If the variable structure for x_i is covered by $5l_i+2$ or more convex polygons, then rearrange the convex polygons in such a way that all beams that point to clause checkers are turned on for positive and negative literals. By construction, this is always possible with $5l_i + 2$ convex polygons.
   - If the variable structure for x_i is covered by $5l_i + 1$ convex polygons and one beam from a beam machine for literal x_i ($\neg x_i$) that is aimed at its associated clause checker is turned on, then rearrange all convex polygons in the variable generator in such a way that all beams from beam machines for literal x_i ($\neg x_i$) that are aimed at the associated clause checkers are turned on.

The convex cover obtained this way is still a feasible solution. After this transformation, we have for each variable structure x_i one of the following cases:

- for all negative and positive literals, the beams that are aimed toward the clause checkers are turned on;
- only for all positive or negative literals, the beams that are aimed toward the clause checkers are turned on;
- for negative and positive literals, the beams that are aimed toward the beam locks are turned on.

We set the truth values for the variables as follows: if all beams of literal x_i ($\neg x_i$) that are aimed at clause checkers and no beams of literal $\neg x_i$ (x_i) that are aimed at clause checkers are turned on, then let the variable x_i have truth value TRUE (FALSE). If either all or no beams (of both literals x_i and $\neg x_i$) that are aimed at clause checkers are turned on, then let variable x_i be TRUE.

By construction, every solution of I' must consist of at least $5l + n + 1$ convex polygons. If we transform a solution of I' with $5l + n + 1 + \epsilon n$ convex polygons as indicated above, we get at most ϵn variable structures in which the beams of all literals (positive and negative) that are aimed at the clause checkers are turned on. By assigning all these variables the value TRUE, we falsify at most five clauses for each variable, since each variable appears at most five times as a literal.

Therefore, we get a solution of I with at least $m - 5\epsilon n$ clauses satisfied. Since $3m \geq n$, the solution has at least $m(1 - 15\epsilon)$ satisfied clauses. □

In the so-called promise problem [1] of MAXIMUM 5-OCCURRENCE-3-SAT as described above, we are promised that either all clauses are satisfiable or at most a fraction of $1 - 15\epsilon$ of the clauses is satisfiable, and we are to find out which of the two possibilities is true. This problem is *NP*-hard for sufficiently small values of $\epsilon > 0$ (see [1]). Therefore, Theorem 4 implies that the promise problem for MINIMUM CONVEX COVER, where we are promised that the minimum solution contains either $5l + n + 1$ convex polygons or at least $5l + n + 1 + \epsilon n$ convex polygons, is *NP*-hard as well for sufficiently small values of $\epsilon > 0$. Therefore, MINIMUM CONVEX COVER cannot be approximated with a ratio of $\frac{5l+n+1+\epsilon n}{5l+n+1} \geq 1 + \frac{\epsilon n}{25n+n+1} \geq 1 + \frac{\epsilon}{27}$, where we have used that $l \leq 5n$ and $n \geq 1$. This establishes the following theorem.

THEOREM 5. MINIMUM CONVEX COVER *on input polygons with or without holes is APX-hard.*

7. Conclusion. We have proposed a polynomial-time approximation algorithm for MINIMUM CONVEX COVER that achieves an approximation ratio that is logarithmic in the number of vertices of the input polygon. This has been achieved by showing that there is a discretized version of the problem using no more than three times the number of cover polygons. The discretization may be a first step toward answering the long-standing open question of whether the decision version of the MINIMUM CONVEX COVER problem is in *NP* [18]: we know now that there always exists an optimum solution such that the convex polygons in such an optimum solution contain only a polynomial number of vertices and that a considerable fraction of these vertices are actually vertices from the input polygon; however, all other vertices of the convex polygons could still need a superpolynomial number of bits for their coordinates to be expressed. Apart from the discretization, our algorithm applies a MINIMUM SET COVER approximation algorithm to a MINIMUM SET COVER instance with an exponential number of sets that are represented only implicitly, through the geometry. We propose an algorithm that picks the best of the implicitly represented sets with a dynamic programming approach and hence runs in polynomial time. This technique may prove to be of interest for other problems as well. Moreover, by showing *APX*-hardness, we have eliminated the possibility of the existence of a polynomial-time approximation scheme for this problem. However, polynomial-time algorithms could still achieve constant approximation ratios. Whether our algorithm is the best asymptotically possible is therefore an open problem. Furthermore, our algorithm has a rather excessive running time of $O(n^{29} \log n)$, and it is by no means clear how this can be improved substantially.

Acknowledgments. We would like to thank anonymous referees for numerous valuable comments that greatly improved this paper; in particular, we thank the referees for pointing us to [11] and [16] and for providing the running time analysis in section 5 that reduces the overall running time of our algorithm from $O(n^{44})$, which we had in earlier versions of this paper, to $O(n^{29} \log n)$.

REFERENCES

[1] S. Arora and C. Lund, *Hardness of approximations*, in Approximation Algorithms for NP-Hard Problems, D. Hochbaum, ed., PWS Publishing Company, Boston, 1996, pp. 399–446.
[2] B. Chazelle and D. P. Dobkin, *Optimal convex decompositions*, in Computational Geometry, Mach. Intelligence Pattern Recogn. 2, North–Holland, Amsterdam, 1985, pp. 63–133.
[3] P. Crescenzi and V. Kann, *A compendium of NP optimization problems*, in Complexity and Approximation. Combinatorial Optimization Problems and Their Approximability Properties, G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, eds., Springer-Verlag, Berlin, 1999, pp. 87–122.
[4] J. C. Culberson and R. A. Reckhow, *Covering polygons is hard*, J. Algorithms, 17 (1994), pp. 2–44.
[5] D. P. Dobkin, H. Edelsbrunner, and M. H. Overmars, *Searching for empty convex polygons*, Algorithmica, 5 (1990), pp. 561–571.
[6] S. Eidenbenz, C. Stamm, and P. Widmayer, *Inapproximability results for guarding polygons and terrains*, Algorithmica, 31 (2001), pp. 79–113.
[7] S. Eidenbenz, *(In-)Approximability of Visibility Problems on Polygons and Terrains*, Ph.D. thesis, Dissertation ETH 13683, Zürich, Switzerland, 2000.
[8] S. Eidenbenz and P. Widmayer, *An approximation algorithm for minimum convex cover with logarithmic performance guarantee*, in Algorithms—ESA 2001 (Århus), Lecture Notes in Comput. Sci. 2161, Springer-Verlag, Berlin, 2001, pp. 333–343.
[9] D. S. Franzblau, *Performance guarantees on a sweep-line heuristic for covering rectilinear polygons with rectangles*, SIAM J. Discrete Math., 2 (1989), pp. 307–321.
[10] S. Ghosh, *Approximation algorithm for art gallery problems*, in Proceedings of the Canadian Information Processing Society Congress, Mississauga, ON, Canada, 1987.
[11] J. Gudmundsson and C. Levcopoulos, *Close approximations of minimum rectangular coverings*, J. Comb. Optim., 3 (1999), pp. 437–452.
[12] D. H. Green, *The decomposition of polygons into convex parts*, in Computational Geometry, Advances in Computer Research 1, F. P. Preparata, ed., JAI Press, London, 1983, pp. 235–259.
[13] D. Hochbaum, *Approximating covering and packing problems: Set cover, vertex cover, independent set, and related problems*, in Approximation Algorithms for NP-Hard Problems, D. Hochbaum, ed., PWS Publishing Company, Boston, 1996, pp. 94–143.
[14] V. S. A. Kumar and H. Ramesh, *Covering rectilinear polygons with axis-parallel rectangles*, in Proceedings of the 31st ACM Symposium on Theory of Computing, ACM, New York, 1999, pp. 445–454.
[15] A. Lingas, *The power of nonrectilinear holes*, in Automata, Languages, and Programming, Lecture Notes in Comput. Sci. 140, Springer-Verlag, Berlin, 1982, pp. 369–383.
[16] A. Lingas and V. Soltan, *Minimum convex partition of a polygon with holes by cuts in given directions*, Theory Comput. Syst., 31 (1998), pp. 507–538.
[17] J. O'Rourke and K. J. Supowit, *Some NP-hard polygon decomposition problems*, IEEE Trans. Inform. Theory, 29 (1983), pp. 181–190.
[18] T. Shermer, *Recent results in art galleries*, Proc. IEEE, 80 (1992), pp. 1384–1399.
[19] J. Urrutia, *Art gallery and illumination problems*, in Handbook on Computational Geometry, J.-R. Sack and J. Urrutia, eds., Elsevier, New York, 2000, pp. 973–1027.